IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

WRITE OPERATION CONTROL IN STORAGE NETWORKS

Inventor:
Robert A. Cochran

ATTORNEY'S DOCKET NO. 200210226-1

WRITE OPERATION CONTROL IN STORAGE NETWORKS

TECHNICAL FIELD

[0001] The described subject matter relates to electronic computing, and more particularly to systems and methods for managing storage in electronic computing systems.

BACKGROUND

[0002] Effective collection, management, and control of information have become a central component of modern business processes. To this end, many businesses, both large and small, now implement computer-based information management systems.

[0003] Data management is an important component of computer-based information management systems. Many users implement storage networks to manage data operations in computer-based information management systems. Storage networks have evolved in computing power and complexity to provide highly reliable, managed storage solutions that may be distributed across a wide geographic area.

[0004] Data redundancy is one aspect of reliability in storage networks. A single copy of data is vulnerable if the network element on which the data resides fails. If the vulnerable data or the network element on which it resides can be recovered, then the loss may be temporary. However, if either the data or the network element cannot be recovered then the vulnerable data may be lost permanently.

[0005] Storage networks implement remote copy procedures to provide data redundancy and failover procedures to provide data consistency in the event of a failure of one or more network elements. Remote copy procedures replicate one or more data sets resident on a first storage site onto at least a second storage site. A data consistency group (DCG) is a data set comprising a plurality of storage units, each containing a portion of an aggregated data set, with each storage unit having the potential to be individually replicated. The storage units may be logical or physical. A DCG implemented at the disk array level enables data sets to be aggregated across multiple logical units with the assurance that any unsuccessful replication will immediately halt all local and remote write operations, such that the aggregated primary data set and the aggregated secondary data set remain consistent, and therefore useful for continuing operations.

[0006] Large storage networks may comprise dozens, or even hundreds of storage cells, and may have hundreds, or even thousands of host computers that execute write operations to data sets in the storage network. Effective storage management techniques must ensure data consistency in DCGs implemented in complex storage networks.

SUMMARY

[0007] In an exemplary implementation a storage network is provided. The storage network comprises a plurality of storage cells, at least one storage cell comprising physical storage media and a storage media controller that controls data transfer operations with the storage media; a plurality of host computers configurable to execute write operations to at least one storage cell; at least one write control server that regulates the write operations of one or more host computers; and a communication network that provides communication connections between the storage cells, the host computers, and the write control server.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Fig. 1 is a schematic illustration of an exemplary implementation of a networked computing system that utilizes a storage network;

[0009] Fig. 2 is a schematic illustration of an exemplary implementation of a storage network;

[0010] Fig. 3 is a schematic illustration of an exemplary implementation of a computing device that can be utilized to implement a host;

[0011] Fig. 4 is a schematic illustration of an exemplary implementation of a storage cell;

[0012] Fig. 5 is a flowchart illustrating operations in a first exemplary implementation for executing write operations in a storage network;

[0013] Fig. 6 is a flowchart illustrating operations in a second exemplary implementation for executing write operations in a storage network; and

[0014] Fig. 7 is a flowchart illustrating operations in a third exemplary implementation for executing write operations in a storage network.

DETAILED DESCRIPTION

[0015] Described herein are exemplary storage network architectures and methods for unified write block ordering in copy operations. The methods described herein may be embodied as logic instructions on a computer-readable medium. When executed on a processor, the logic instructions cause a general purpose computing device to be programmed as a special-purpose machine that implements the described methods.

Exemplary Network Architecture

[0016] **Fig. 1** is a schematic illustration of an exemplary implementation of a networked computing system 100 that utilizes a storage network. The storage network comprises a storage pool 110, which comprises an arbitrarily large quantity of storage space. In practice, a storage pool 110 has a finite size limit determined by the particular hardware used to implement the storage pool 110. However, there are few theoretical limits to the storage space available in a storage pool 110.

[0017] A plurality of logical disks (also called logical units or LUs) 112a, 112b may be allocated within storage pool 110. Each LU 112a, 112b comprises a contiguous range of logical addresses that can be addressed by host devices 120, 122, 124 and 128 by mapping requests from the connection protocol used by the host device to the uniquely identified LU 102. As used herein, the term "host" comprises a computing system(s) that utilize storage on its own behalf, or on behalf of systems coupled to the host. For example, a host may be a supercomputer processing large databases or a transaction processing server maintaining transaction records. Alternatively, a host may be a file server on a local area network (LAN) or wide area network (WAN) that provides storage services for an enterprise. A file server may comprise one or more disk controllers and/or RAID controllers configured to manage multiple disk drives. A host connects to a storage network via a communication connection such as, e.g., a Fibre Channel (FC) connection.

[0018] A host such as server 128 may provide services to other computing or data processing systems or devices. For example, client computer 126 may access storage pool 110 via a host such as server 128. Server 128 may provide file services to client 126, and may provide other services such as transaction processing services, email services, etc. Hence, client device 126 may or may not directly use the storage consumed by host 128.

[0019] Devices such as wireless device 120, and computers 122, 124, which are also hosts, may logically couple directly to LUs 112a, 112b. Hosts

120-128 may couple to multiple LUs 112a, 112b, and LUs 112a, 112b may be shared among multiple hosts. Each of the devices shown in FIG. 1 may include memory, mass storage, and a degree of data processing capability sufficient to manage a network connection.

[0020] **Fig. 2** is a schematic illustration of an exemplary storage network 200 that may be used to implement a storage pool such as storage pool 110. Storage network 200 comprises a plurality of storage cells 210a, 210b, 210c connected by a communication network 212. Storage cells 210a, 210b, 210c may be implemented as one or more communicatively connected storage devices. Exemplary storage devices include the STORAGEWORKS line of storage devices commercially available form Hewlett-Packard Corporation of Palo Alto, California, USA. Communication network 212 may be implemented as a private, dedicated network such as, e.g., a Fibre Channel (FC) switching fabric. Alternatively, portions of communication network 212 may be implemented using public communication networks pursuant to a suitable communication protocol such as, e.g., the Internet Small Computer Serial Interface (iSCSI) protocol.

[0021] Client computers 214a, 214b, 214c may access storage cells 210a, 210b, 210c through a host, such as servers 216, 220. Clients 241a, 214b, 214c may be connected to file server 216 directly, or via a network 218 such as a Local Area Network (LAN) or a Wide Area Network (WAN). The number of storage cells 210a, 210b, 210c that can be included in any storage network is

limited primarily by the connectivity implemented in the communication network 212. A switching fabric comprising a single FC switch can interconnect 256 or more ports, providing a possibility of hundreds of storage cells 210a, 210b, 210c in a single storage network.

[0022] Hundreds or even thousands of host computers may connect to storage network 200 to access data stored in storage cells 210a, 210b, 210c. Storage network 200 further comprises at least one write control server 230 that regulates write operations of host computers that connect to storage network 200. Operation of the write control server is explained in detail below.

[0023] Hosts 216, 220 and write control server 230 may be embodied as server computers. **Fig. 3** is a schematic illustration of an exemplary computing device 330 that can be utilized to implement a host. Computing device 330 includes one or more processors or processing units 332, a system memory 334, and a bus 336 that couples various system components including the system memory 334 to processors 332. The bus 336 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. The system memory 334 includes read only memory (ROM) 338 and random access memory (RAM) 340. A basic input/output system (BIOS) 342, containing the basic routines that help to transfer information between elements within computing device 330, such as during start-up, is stored in ROM 338.

[0024] Computing device 330 further includes a hard disk drive 344 for reading from and writing to a hard disk (not shown), and may include a magnetic disk drive 346 for reading from and writing to a removable magnetic disk 348, and an optical disk drive 350 for reading from or writing to a removable optical disk 352 such as a CD ROM or other optical media. The hard disk drive 344, magnetic disk drive 346, and optical disk drive 350 are connected to the bus 336 by a SCSI interface 354 or some other appropriate interface. The drives and their associated computer-readable media provide nonvolatile storage of computer-readable instructions, data structures, program modules and other data for computing device 330. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 348 and a removable optical disk 352, other types of computer-readable media such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs), read only memories (ROMs), and the like, may also be used in the exemplary operating environment.

[0025] A number of program modules may be stored on the hard disk 344, magnetic disk 348, optical disk 352, ROM 338, or RAM 340, including an operating system 358, one or more application programs 360, other program modules 362, and program data 364. A user may enter commands and information into computing device 330 through input devices such as a keyboard 366 and a pointing device 368. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like.

These and other input devices are connected to the processing unit 332 through an interface 370 that is coupled to the bus 336. A monitor 372 or other type of display device is also connected to the bus 336 via an interface, such as a video adapter 374.

[0026] Computing device 330 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 376. The remote computer 376 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computing device 330, although only a memory storage device 378 has been illustrated in Fig. 3. The logical connections depicted in Fig. 3 include a LAN 380 and a WAN 382.

[0027] When used in a LAN networking environment, computing device 330 is connected to the local network 380 through a network interface or adapter 384. When used in a WAN networking environment, computing device 330 typically includes a modem 386 or other means for establishing communications over the wide area network 382, such as the Internet. The modem 386, which may be internal or external, is connected to the bus 336 via a serial port interface 356. In a networked environment, program modules depicted relative to the computing device 330, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the

network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0028] Hosts 216, 220 may include host adapter hardware and software to enable a connection to communication network 212. The connection to communication network 212 may be through an optical coupling or more conventional conductive cabling depending on the bandwidth requirements. A host adapter may be implemented as a plug-in card on computing device 330. Hosts 216, 220 may implement any number of host adapters to provide as many connections to communication network 212 as the hardware and software support.

[0029] Generally, the data processors of computing device 330 are programmed by means of instructions stored at different times in the various computer-readable storage media of the computer. Programs and operating systems may distributed, for example, on floppy disks, CD-ROMs, or electronically, and are installed or loaded into the secondary memory of a computer. At execution, the programs are loaded at least partially into the computer's primary electronic memory.

[0030] **Fig. 4** is a schematic illustration of an exemplary implementation of a storage cell 400, such as storage cell 210. Referring to Fig. 4, storage cell 400 includes two Network Storage Controllers (NSCs), also referred to as disk controllers, 410a, 410b to manage the operations and the transfer of data to and from one or more disk arrays 440, 442. NSCs 410a, 410b may be implemented

as plug-in cards having a microprocessor 416a, 416b, and memory 418a, 418b. Each NSC 410a, 410b includes dual host adapter ports 412a, 414a, 412b, 414b that provide an interface to a host, i.e., through a communication network such as a switching fabric. In a Fibre Channel implementation, host adapter ports 412a, 412b, 414a, 414b may be implemented as FC N_Ports. Each host adapter port 412a, 412b, 414a, 414b manages the login and interface with a switching fabric, and is assigned a fabric-unique port ID in the login process. The architecture illustrated in Fig. 4 provides a fully-redundant storage cell; only a single NSC is required to implement a storage cell 210. [

[0031] Each NSC 410a, 410b further includes a communication port 428a, 428b that enables a communication connection 438 between the NSCs 410a, 410b. The communication connection 438 may be implemented as a FC point-to-point connection, or pursuant to any other suitable communication protocol.

[0032] In an exemplary implementation, NSCs 410a, 410b further include a plurality of Fiber Channel Arbitrated Loop (FCAL) ports 420a-426a, 420b-426b that implement an FCAL communication connection with a plurality of storage devices, e.g., arrays of disk drives 440, 442. While the illustrated embodiment implement FCAL connections with the arrays of disk drives 440, 442, it will be understood that the communication connection with arrays of disk drives 440, 442 may be implemented using other communication

protocols. For example, rather than an FCAL configuration, a FC switching fabric may be used.

Exemplary Operations

[0033] Having described various components of an exemplary storage network, attention is now directed to operations of the storage network 200 and components thereof.

[0034] In operation, storage capacity provided by the arrays of disk drives 440, 442 in a storage cells 210a, 210b, 210c may be added to the storage pool 110. When an application requires storage capacity, logic instructions on a host computer 128 may establish a LU from storage capacity available on the arrays of disk drives 440, 442 available in one or more storage cells 210a, 210b, 210c. It will be appreciated that because a LU is a logical unit, not a physical unit, the physical storage space that constitutes the LU may be distributed across multiple storage cells 210a, 210b, 210c. Data for the application may be stored on one or more LUs in the storage network.

[0035] Storage network 200 may implement remote copy procedures to provide data redundancy for data stored in storage cells 210a, 210b, 210c. By way of example, referring to Fig. 2, a LU resident on storage cell 210a may have a remote copy resident on storage cell 210b, which may be located at an arbitrary distance from storage cell 210a. Additional remote copies of the LU resident on storage cell 210a may be maintained on other storage cells in the

storage network 210. Similarly, a LU resident on storage cell 210b may have remote copies on storage cell 210a or 210c, and a LU resident on storage cell 210c may have remote copies on storage cell 210a or 210b. During the remote copy process the information in the LU is transmitted across the switching fabric, sometimes referred to as a "network cloud" to its destination storage cell.

[0036] An application that needs access to data in the storage network may launch a read query to a host computer. In response to a read query, the host computer queries the NSC(s) on one or more storage cells in which the requested data resides. The NSC(s) retrieve the requested data from the storage media on which it resides and forwards the data to host computer, which in turn can forward to data to the requesting device.

Write Operations

[0037] An application can write data to the storage network 200 by launching a write request to a host computer 216, 220. In response to a write request, a host computer 216, 220 launches a write command to the NSC(s) 410a, 410b in one or more storage cells 210a, 210b, 210c on which the requested data resides. The write command includes the data to be written to the storage network 200. In response to the write command, the NSC(s) 410a, 410b write the data onto the storage media. If the storage network 200 is configured to implement one or more remote copies of stored data, then data

from the write operation is contemporaneously written to a second storage cell 210a, 210b, 210c on the storage network. The remote copy operation may be implemented by either a host computer 216, 220 or by the NSC(a) 410a, 410b.

[0038] In an exemplary implementation, write operations for a data consistency group in storage network 210 are regulated by write control server 230. Fig. 5 is a flowchart illustrating operations 500 in a first exemplary implementation for executing write operations for a data consistency group in storage network 200. In the exemplary implementation illustrated in Fig. 5, write control server 230 restricts write operations for the data consistency group in storage network 200 to permit only a single host computer 216, 220 to execute a write operation at any point in time.

[0039] At operation 510 a host computer initiates a write request, which is transmitted to write control server 230, e.g., over communication network 212. Write control server 230 receives the write request and, at operation 514, positions the write request in a write permission queue (WPQ), which comprises a list of write requests received from host computers 216, 220 that connect to storage network 200.

[0040] In one exemplary embodiment, writer control server 230 arranges write requests in the WPQ in the order in which the write requests were received at the write control server 230. In alternate embodiments, write control server 230 attempts to arrange the WPQ in accordance with the precise time sequence in which the write requests were generated by host computers

216, 220. This may be accomplished using one of multiple exemplary implementations.

[0041] One exemplary implementation utilizes a reverse handicapping process in which incoming write requests are delayed at the write control server to compensate for transmission delays between a host computer 216, 220 that generated the write request and the write control server 230. The write control server 230 may construct an array that contains the transmission delays associated with host computers 216, 220 connected to the storage network 200. The transmission delay values in the array may be used to compensate for transmission delays when positioning a write request in the WPQ.

[0042] The array may be constructed in a background process executed by write control server 230. To construct the array, write control server 230 pings one or more host computers 216, 220 that connect to storage network 200. The host computers 216, 220 respond to the ping. The round trip time (RTT) in the communication path between write control server 230 and the respective host computers 216, 220 is measured. The RTT is divided by two to approximate the transmission delay, or latency, a write request incurs during transmission from the respective host computers 216, 220 to write control server 230. The transmission delays may be stored in a suitable data structure, e.g., an array stored in memory, and logically associated with their respective host computers 216, 220.

[0043] To compensate for transmission delays, incoming write requests are delayed at the write control server by a time period equal to the longest delay in the array of transmission delays minus the delay associated with the host computer 216, 220 that originated the request. By way of example, assume that host computer 216 exhibits a transmission delay of 20 nanoseconds (ns), which is the longest transmission delay of all host computers connected to storage network 200. A write request from host computer 216 is not delayed at write control server 230 (because 20ns-20ns results in a delay of zero ns). By contrast, assume that host computer 220 exhibits a transmission delay of 14 ns. A write request from host computer 220 will be delayed by six ns (20ns-14ns). Thus, the reverse handicapping process attempts to compensate for transmission delays between a host computer 216, 220 and the write control server 230 when positioning write requests in the WPQ.

[0044] The process of constructing an array of transmission delays may be performed periodically as a background process, or in response to a write requests from host computers 216, 220. Write control server 230 may ping the host computers 216, 220 once, or multiple times, to derive an average RTT.

[0045] In alternate implementations, write requests from host computers 216, 220 are marked with a time stamp indicating a time at which the write request was originated. If the clocks in the host computers 216, 220 are synchronized with the clock in the write control server 230, then the write requests may be positioned in the WPQ in accordance with their respective time

stamps. By contrast, if the clocks in the host computers 216, 220 are not synchronized with the clock in the write control server 230, then the transmission delay may be used to position the write requests in the WPQ, as described above.

[0046] Referring again to Fig. 5, at operation 518 write control server 230 generates a write access token, which is transmitted to the host computer 216, 220 at the top of the WPQ. The write access token may be embodied as a message from write control server 230, which may include information identifying the write request from the host computer 216, 220 such as, e.g., a sequence number or a time stamp at which the host computer generated the write request. The token may also include a time stamp indicating a time at which write permission was granted by the write control server 230.

[0047] The write access token is received by the host computer 216, 220 and at operation 522 the host computer 216, 220 initiates the write operation associated with the write access token. It will be appreciated that a host computer 216, 220 may have multiple outstanding write requests in the WPQ. The write operation may be associated with the write request by matching the sequence number or time stamp returned with the write access token. The write operation may be performed in accordance with normal storage network operations. If the storage network is configured to generate a redundant copy of the data, then redundant copy operations are also executed.

[0048] Operations 526 and 530 are optional. At operation 526 a host computer 216, 220 generates a write initiation signal, which is transmitted to write control server 230. The write initiation signal includes information that identifies the write operation and may include a time stamp indicating a time at which the write operation began. At operation 530 write control server 230 stores the write initiation signal in a suitable storage location, e.g., in a memory communicatively connected to write control server 230.

[0049] At operation 534 host computer 216, 220 generates a write completion signal when write operations, including any redundant copy operations, are successfully completed or result in failure. The write completion signal is transmitted to write control server 230. If the write completion signal indicates success (operation 538), then write control server 230 transmits a write access token to the host computer 216, 220 associated with the next write request in the WPQ. By contrast, if the write completion signal indicates failure, then all write operations for the data consistency group in the storage network are terminated (operation 542). Write control server 230 may then transmit a write failure signal (operation 546) that triggers failover procedures for the data consistency group in the storage network.

[0050] The operations 510-538 permit the write control server 230 to regulate write operations for the data consistency group in the storage network 200 such that only one host computer 216, 220 has write access at any given point in time. Upon successful completion of a write operation by a host

computer, e.g., 216, write access is granted to another host computer, e.g., 220. By contrast, failure of a write operation terminates all write operations for the data consistency group in the storage network. Thus, data consistency is always maintained for the data sets in the data consistency group as well as their local or remote replicas.

[0051] In an optional feature the write control server 230 may implement a timeout procedure that terminates write operations if a write operation exceeds a time threshold. By way of example, write control server 230 may initiate a timer when the write initiation signal is received in operation 530. If a write completion signal is not received within a predetermined time period, then the write control server may generate a signal that triggers failover procedures for the data consistency group in the storage network.

[0052] It will be appreciated that incoming write requests may be received at the write control server 230 and positioned in the WPQ while host computers 216, 220 are executing a write operation. In that regard, operations 510 and 514 may be executed repeatedly independent of the remaining operations illustrated in Fig. 5.

[0053] **Fig. 6** is a flowchart illustrating operations in a second exemplary implementation for executing write operations in a storage network 200. In the exemplary implementation illustrated in Fig. 6, write control server 230 authorizes write operations for the data consistency group in storage network 200, permits multiple host computers 216, 220 to execute write operations at

any point in time, and maintains a response log tracking write operations in the data consistency group. In the event of a write failure, write control server 230 consults the response log to determine a time at which the failed write was authorized and transmits a write failure signal to host computers identified in the response log. In response to the write failure signal, the host computers can undo writes that occurred after the write failure, thus ensuring a consistent data set in the local and replicated data consistency group.

[0054] Referring to Fig. 6, at operation 610 a host computer initiates a write request, which is transmitted to write control server 230, e.g., over communication network 212. Write control server 230 receives the write request at operation 632 and, at operation 634, positions the write request in the WPQ. Write control server 230 may implement one of the same procedures for positioning received write requests in the WPQ as described in connection with Fig. 5.

[0055] At operation 640 write control server 230 generates a write access token, which is transmitted to the host computer 216, 220 at the top of the WPQ. The write access token may be embodied as a message from write control server 230, which may include information identifying the write request from the host computer 216, 220 such as, e.g., a sequence number or a time stamp at which the host computer generated the write request. The token may also include a time stamp indicating a time at which write permission was granted by the write control server 230. The write access token is transmitted

to the requesting host computer 216, 220, and entered into a response log maintained by write control server 230. Entries in the response log comprise an identifier associated with the host computer 216, 220 to which write permission was granted and a time stamp identifying the time at which write permission was granted. The response log may be maintained in a suitable data structure, e.g., an array or a linked list, in a memory location communicatively connected to write control server 230 such as, e.g., the RAM or disk memory of write control server.

[0056] The write access token is received by the host computer 216, 220 and at operation 614 the host computer 216, 220 initiates the write operation associated with the write access token. The write operation may comprise remote copy operations as described in connection with Fig. 5. It will be appreciated that a host computer 216, 220 may have multiple outstanding write requests in the WPQ. The write operation may be associated with the write request by matching the sequence number or time stamp returned with the write access token. The write operation may be performed in accordance with normal storage network operations. If the storage network is configured to generate a redundant copy of the data, then redundant copy operations are also executed.

[0057] In an exemplary embodiment each host computer maintains an undo log that comprises information required to undo a write operation. An undo log for storage network 200 may comprise entries for the time stamp

transmitted with the write access token, a time at which the write operation begins, a time at which the write operation concludes, an address identifying a location at which the contents of the write operation are stored, the contents of the write operation, and the status of the write operation. The undo log should be large enough to enable the host to undo writes that occurred in a time period corresponding to the longest transmission delay between write control server 230 and a host computer 216, 220 in the data consistency group plus the longest write timeout implemented in the data consistency group. At operation 618 the host computer 216, 220 enters the write operation into its undo log.

[0058] At operation 622 the host computer 216, 220 generates a write operation status signal, which is transmitted to write control server 230. The write control status signal comprises one or more entries that identify the write operation and an entry that indicates whether the write operation completed successfully or failed. In the event of a failure, the write operation status signal functions as a write failure alarm.

[0059] If, at step 624 the host computer 216, 220 determines that the write operation was completed successfully, then the host computer continues normal operations. By contrast, if the write operation failed, then the host computer 216, 220 stops all write operations at operation 626 and consults the undo log to undo any write operations that occurred after the time that permission was granted for the failed write operation (operation 628).

[0060] The write operation status signal generated in operation 622 is received at write control server 230, which analyzes the write operation status signal to determine whether it was a success or failure (operation 644). If the write operation completed successfully, then write control server 230 continues normal operations. By contrast, if the write operation failed, then write control server terminates granting permission for write operations (operation 648) and transmits a write failure signal to the host computers 216, 220 in the response log (operation 652). The write failure signal comprises a time stamp identifying the time at which write control server granted permission for the failed write operation.

[0061] Upon receipt of the write failure signal, the host computers 216, 220 stop all write operations and consult their undo log to undo any write operations that occurred after the time that permission was granted for the failed write operation, i.e., the host computers execute operations 626 and 628. It will be appreciated that the host computer that experienced the failed write may already have executed operations 626 and 628 based on its own failure information.

[0062] The operations illustrated in Fig. 6 enable multiple host computers 216, 220 in the data consistency group to execute write operations at the same time, which provides better performance than the operations illustrated in Fig. 5. In the event of a failed write operation, the local and replica data consistency group is restored to a consistent data state.

[0063] In another exemplary implementation the operations illustrated in Figs. 5 and 6 may be modified slightly to accommodate the introduction of a Uniform Time Broadcaster (UTB) component into the storage network. The UTB component broadcasts a timing signal that may be used by all components in storage network 200. The UTB component may be implemented in write control server 230, e.g., by using the internet Small Computer Serial Interface (iSCSI) broadcast transmission mode, or as a separate component. The timing signal may be broadcast over the communication network 212, or over a separate communication network. The UTB may also be implemented as a receiver of GPS satellite time.

[0064] Introduction of a UTB permits write control server 230 to implement a more deterministic process of sorting write requests from host computers 216, 220 in the WPQ. Because the storage network uses uniform timing, write requests may be supplied with a unified time stamp from the originating host, which may be positioned in the WPQ in accordance with the unified time stamp indicating the time at which the write request was generated. Accordingly, a reverse handicapping procedure is not necessary. By contrast, if the write request does not include a unified time stamp from the originating host, then the write control server 230 may optionally implement a reverse handicapping procedure to compensate for transmission delays between the host computer that generated the write request and the write control server. The remaining operations may be implemented as described in Figs. 5 and 6.

[0065] In another exemplary implementation the UTB cooperates with the host computers 216, 220 to eliminate the need for write control server 230 to authorize or track write operations for the data consistency group.

[0066] Fig. 7 is a flowchart illustrating operations in a third exemplary implementation for executing write operations in a storage network 200. The operations illustrated in Fig. 7 may be implemented in a processor associated with a host computer in storage network 200.

[0067] At operation 710 a host computer 216, 220 receives a signal comprising a timing indicator from the UTB. In an exemplary implementation the UTB constantly broadcasts a timing signal to the host computers in storage network 200, so the host computer 216, 220 constantly receives a clock signal representing the clock signal of the UTB. In alternate embodiments the UTB may periodically transmit a timing signal and the host computers 216, 220 may synchronize their clocks with the clock signal from the UTB or maintain a signal that represents the difference between the UTB timing indicator and a local timer. Thus, the host computers maintain timing information that identifies the timing signal of the UTB. The timing information may be the timing indicator from the UTB, a synchronized local timer, or a signal that represents the difference between the UTB timing indicator and a local timer.

[0068] Optionally, the host computer 216, 220 may maintain information about the transmission delay between the UTB and the host computer and may record this information in association with the timing information. For

example, the host computer may periodically ping the UTB and determine the RTT for a response from the UTB. The RTT may be divided by two to estimate the transmission delay.

[0069] At operation 714 the host computer initiates a write operation, e.g., in response to a write request from a client. At operation 718 the write operation is entered into an undo log. In an exemplary embodiment each host computer maintains an undo log that comprises information required to undo a write operation. The undo log may be implemented substantially as described in connection with Fig. 6. The host computer 216, 220 records in the undo log the timing information that identifies the timing signal of the UTB at the time when the write operation was initiated. This provides an association between the UTB timing indicator and the write operation. The timing information may include the transmission delay.

[0070] If the write operation is completed successfully (operation 722), then the host computer 216, 220 continues normal operations, so control may pass back to operation 710. By contrast, if the write operation fails, then control passes to operation 726, and the host computer 216, 220 transmits a write failure signal to other host computers 216, 220 in the storage network 200. The write failure signal includes the timing information that identifies the UTB timing indicator associated with the failed write operation.

[0071] In an exemplary implementation the host computer 216, 220 broadcasts the signal directly to other host computers 216, 220 in the storage

network, e.g., using the iSCSI broadcast transmission mode. In an alternate implementation, the host computer transmits the write failure signal to write control server 230, which transmits the write failure signal to other host computers 216, 220 in storage network 200.

[0072] In response to the write failure signal, at operation 730, the host computers 216, 220 stop all write operations, and at operation 734 the host computers 216, 220 use their respective undo logs to undo any write operations that were initiated after the failed write operation. It will be appreciated that the host computer 216, 220 that originate the write failure signal may execute operations 730 and 734 in response to the local write failure signal.

[0073] The operations illustrated in Fig. 7 enable multiple host computers 216, 220 in the data consistency group to execute write operations at the same time, and without central control of write access. In the event of a failed write operation, the data consistency group is restored to a consistent data state.

[0074] In addition to the specific embodiments explicitly set forth herein, other aspects and embodiments of the present invention will be apparent to those skilled in the art from consideration of the specification disclosed herein. It is intended that the specification and illustrated embodiments be considered as examples only, with a true scope and spirit of the invention being indicated by the following claims.